UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/662,293	09/16/2003	Michel Doyon	10442-30US	9796

20988 7590 06/07/2011

NORTON ROSE OR LLP
1, Place Ville Marie
SUITE 2500
MONTREAL, QC H3B 1R1
CANADA

EXAMINER	
VERDI, KIMBLEANN C	
ART UNIT	PAPER NUMBER
2194	

MAIL DATE	DELIVERY MODE
06/07/2011	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/662,293	DOYON ET AL.	
	Examiner	Art Unit	
	KIMBLEANN VERDI	2194	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *04 April 2011*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-14* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-14* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1 – 14 are pending in the application.

Continued Examination Under 37 CFR 1.114

2. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on April 4, 2011 has been entered.

Claim Objections

3. Claim 1 is objected to because of the following informalities: line 3, the recitation of "CPU" should be preceded by the full term it represents. Appropriate correction is required.

Claim Rejections - 35 USC § 112

4. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

5. **Claims 4-6 and 8-10 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.**

6. Claim 4 recites the limitation "said sequence of asynchronous commands" in line 2. There is insufficient antecedent basis for this limitation in the claim. For purposes of examination the limitation of “said sequence of asynchronous commands” is interpreted as one of the plurality of sequences of asynchronous commands as recited in claim 2.

7. Claim 8 recites the limitation "said executing said commands" in line 1. There is insufficient antecedent basis for this limitation in the claim. Claim 9 does not cure the deficiencies of claim 8. For purposes of examination the limitation of “said executing said commands” is interpreted as said initiating said commands as recited in claim 1.

8. Claims 5 and 10 contain the trademark/trade name WINDOWS™. Where a trademark or trade name is used in a claim as a limitation to identify or describe a particular material or product, the claim does not comply with the requirements of 35 U.S.C. 112, second paragraph. See *Ex parte Simpson*, 218 USPQ 1020 (Bd. App. 1982). The claim scope is uncertain since the trademark or trade name cannot be used properly to identify any particular material or product. A trademark or trade name is used to identify a source of goods, and not the goods themselves. Thus, a trademark or trade name does not identify or describe the goods associated with the trademark or trade name. In the present case, the trademark/trade name is used to identify/describe a non real time operating system and, accordingly, the identification/description is indefinite. Claim 6 does not cure the deficiencies of claim 5.

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. **Claims 1- 2 are rejected under 35 U.S.C. 103(a) as being unpatentable over Baertsch et al. (hereinafter Baertsch, previously cited) (U.S. Patent 6,470,071 B1) in view of Wilt et al. (hereinafter Wilt) (U.S. Patent No. 7,234,144 B2), and further in view of Nabekura et al. (hereinafter Nabekura, previously cited on PTO-892 mailed on 4/27/2009) (U.S. Patent No. 5,530,815).**

11. **As to claim 1**, Baertsch teaches the invention substantially as claimed including a method for providing improved real time command execution in a non real time operating system, comprising:

executing at least one application (***i.e. "User App. 301", Figure 15***) at user level mode (***i.e. -User App. 301 is a user interface which executes at user mode as shown in Figure 71-, "As illustrated, interface 730 includes a plurality of user interfaces 732, which interfaces with operating system kernel 734", col. 72, lines 54-56***) from at least one CPU running the non real time operating system (***i.e. "System 300 includes host computer 114 running user application 301 from script 309",***

col. 12, lines 18-19, "on host computer 114 running a non-real time operating system to support an event compiler", col. 14, line 34);

having said at least one application (***i.e. "User App. 301", Figure 15***) at said user mode level (***i.e. –User App. 301 is a user interface which executes at user mode as shown in Figure 71-, "As illustrated, interface 730 includes a plurality of user interfaces 732, which interfaces with operating system kernel 734", col. 72, lines 54-56***) determine a sequence to be followed for a set of commands (***i.e. "frame sequence 310", "An exact sequence of image frames and associated acquisition parameters is needed in advance for a particular image acquisition. Accordingly, one can specify, for each frame, the readout delay relative to x-ray pulse, the detector parameters, etc. A description of such attributes is captured in a frame sequence 310 of script 309. Program applications configure the data acquisition system through the frame sequence structure and then trigger the system to initiate acquisition of the frames", col. 14, lines 10-18)***;

providing (***i.e. creating and sending***) from said at least one application (***"User App. 301", Figure 15***) said sequence of commands (***i.e. "FIG. 15 is a block diagram showing the flow of control information and data through system 300 during image acquisition. As illustrated, frame sequence 310 is created by way of script 309", col. 14, lines 39-42***) to a privileged mode (***i.e. DFN device driver 314 operates at kernel mode, Figure 71***) of said non real time operating system (***i.e. "Frame sequence 310 is then translated into event sequence 312 using a compiler, which knows the details of the target control hardware. Event sequence 312 is received***

by test control unit 311, then sent to DFN device driver 314 ", col. 14, lines 42-46) to be executed in real time (***i.e. "Once the event sequence 312 is known, the details are transmitted to DFN 304 for execution in real-time", col. 14, lines 36-38, "Event sequence 312 is received by test control unit 311, then sent to DFN device driver 314, over computer communication bus 302, and finally to detector framing node 304. The event sequence 312 is then stored in preparation for execution", col. 14, lines 44-48***).

12. Baertsch does not explicitly disclose storing said sequence of commands in a command queue to be accessible from the privileged mode; and initiating one at a time, using the at least one CPU of each of said commands from said stored sequence of commands.

13. However Wilt teaches storing said sequence of commands (***i.e. "stream of commands"***) in a command queue (***i.e. "command buffer 315, Figure 3B, "FIG. 3B shows a variant in which the runtime 312 emits a hardware-independent stream of commands into a buffer 313, which is then parsed by the driver 314 and written into a command buffer 315", col. 17, lines 21-24***) to be accessible from the privileged mode (***i.e. "When the DP2 token stream is submitted, a kernel transition occurs and the driver 314 translates the DP2 token stream into hardware-specific commands in kernel mode. FIG. 3B does not make any assumptions about whether the driver 314 is in user mode or kernel mode, and in this regard, driver***

component 314 could reside in user mode as well as kernel mode", col. 17, lines 29-34).

14. Baertsch as modified by Wilt does not explicitly disclose initiating one at a time, using the at least one CPU of each of said commands from said stored sequence of commands.

15. However Nabekura teaches initiating one at a time, using the at least one CPU (***i.e. "Processor Unit 300", Figure 3***) of each of said commands from said stored sequence of commands (***i.e. "The command selector 16 fetches the command which can be executed from the commands held as a queue in the command queue 14 and supplies the command to the asynchronous pipeline computing unit 18", col. 4, lines 46-50***).

16. It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified the invention of Baertsch to incorporate the teachings of Wilt and Nabekura. One of ordinary skill in the art would have been motivated to make the combination because these features would have provided a mechanism which enables multiple applications to efficiently share the computational resources available in the system (***col. 3, lines 33-35 of Wilt***) and a mechanism in which the order and operation are guaranteed when asynchronous commands held in a command queue are executed (***col. 2, lines 16-18 of Nabekura***).

17. **As to claim 2**, Baertsch teaches the invention substantially as claimed including wherein a plurality of sequences of asynchronous commands is provided (**i.e.** ***"Frame sequence 310 is then translated into event sequence 312 using a compiler, which knows the details of the target control hardware", col. 14, lines 42-44***), each sequence being related to a corresponding application thread (***i.e. "As illustrated, interface 730 includes a plurality of user interfaces 732, which interfaces with operating system kernel 734", col. 72, lines 54-56 "FIG. 15 is a block diagram showing the flow of control information and data through system 300 during image acquisition. As illustrated, frame sequence 310 is created by way of script 309", col. 14, lines 39-42***).

18. Baertsch does not explicitly disclose said storing of a sequence of commands is performed in a corresponding queue from the execution of said corresponding application thread queue.

19. However Wilt teaches said storing of a sequence of commands is performed in a corresponding queue (***i.e. "command buffer 315***) from the execution of said corresponding application thread queue (***i.e. "buffer 313", "command buffer 315, Figure 3B, "FIG. 3B shows a variant in which the runtime 312 emits a hardware-independent stream of commands into a buffer 313, which is then parsed by the driver 314 and written into a command buffer 315", col. 17, lines 21-24***)).

20. The motivation for modifying Baertsch with the teachings of Wilt and Nabekura is the same as provided in the rejection of claim 1 above.

21. **Claims 3 – 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Baertsch et al. (hereinafter Baertsch, previously cited) (U.S. Patent 6,470,071 B1) in view of Wilt et al. (hereinafter Wilt) (U.S. Patent No. 7,234,144 B2), and further in view of Nabekura et al. (hereinafter Nabekura, previously cited on PTO-892 mailed on 4/27/2009) (U.S. Patent No. 5,530,815), as applied to claims 1 and 2 above, and further in view of Dingwall et al. (hereinafter Dingwall, previously cited) (U.S. Patent No. 5,903,752).**

22. **As to claim 3**, Baertsch as modified by Wilt and Nabekura does not explicitly disclose wherein a synchronous command is added to said sequence of commands, said at least one application sleeping until said synchronous command is executed.

23. However Dingwall teaches wherein a synchronous (***i.e. real-time***) command is added to said sequence of commands, said at least one application sleeping (***i.e. application task is asleep (dormant/locked) until interrupted, 818, Fig. 8***) until said synchronous command is executed (***i.e. RT Scheduler 30, Fig. 2, releases scheduling lock which allows real-time tasks to pre-empt the current (asynchronous) process, col. 3, lines 59-61***).

24. It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have further modified the invention of Baertsch as modified by Wilt and Nabekura to incorporate the features of Dingwall. One of ordinary skill in the art would have been motivated to make the combination because this allows real-time programming with support for the presentation of natural data types, without allowing other operations to disrupt the delivery and playback of the audio and video data (***col. 1, lines 66-67 and col. 2, lines 1-3 of Dingwall***).

25. **As to claim 4**, Baertsch as modified by Wilt and Nabekura does not explicitly disclose wherein a synchronous command is added to said sequence of asynchronous commands, said corresponding application thread sleeping until said synchronous command is executed.

26. However Dingwall teaches wherein a synchronous command is added to said sequence of asynchronous commands, said corresponding application thread sleeping (***i.e. application task is asleep (dormant/locked) until interrupted, 818, Fig. 8***) until said synchronous command is executed (***i.e. RT Scheduler 30, Fig. 2, releases scheduling lock which allows real-time tasks to pre-empt the current (asynchronous) process, col. 3, lines 59-61***).

27. The motivation for further modifying Baertsch with the teachings of Wilt, Nabekura, and Dingwall is the same as provided in the rejection of claim 3 above.

28. **As to claim 5**, Baertsch as modified by Wilt and Nabekura does not explicitly disclose wherein said non real time operating system is MICROSOFT WINDOWS™ and said storing said sequence of commands is performed through execution of a driver routine from a DLL file.

29. However Dingwall teaches wherein said non real time operating system is MICROSOFT WINDOWS™ (***i.e. environment of WINDOWS™, col. 3, lines 33-34)*** and said storing said sequence of commands is performed through execution of a driver routine from a DLL file ***(Virtual Device Driver (VxD) is dynamic link library (DLL), col. 3, lines 33-36***).

30. The motivation for further modifying Baertsch with the teachings of Wilt, Nabekura, and Dingwall is the same as provided in the rejection of claim 3 above.

31. **As to claim 6**, Baertsch does not explicitly disclose wherein said providing said sequence of commands involves said commands being pushed one at a time into said sequence through system call.

32. However Dingwall teaches wherein said providing sequence of commands involves said commands being pushed one at a time into said sequence through system call (***i.e. interrupt occurs which causes the processor to switch to VxD interrupt***

mode and execute RT interrupt handler 32, Fig. 2, col. 4, lines 51-23, RT interrupt handler 32, Fig. 2, wake up associated real-time task).

33. The motivation for further modifying Baertsch with the teachings of Wilt, Nabekura, and Dingwall is the same as provided in the rejection of claim 3 above.

34. **As to claim 7**, Baertsch as modified by Wilt and Nabekura does not explicitly disclose wherein at least one of said stored commands is a branch command to control the order of execution of said stored commands.

35. However Dingwall teaches wherein at least one of said stored commands is a branch command to control the order of execution of said stored commands (***i.e. RT scheduler 30, Fig. 2, schedules task preemptively by priority and allows interrupt handlers 32, Fig. 2, to make real-time tasks ready for execution without preemption, col. 3, lines 54-62***).

36. The motivation for further modifying Baertsch with the teachings of Wilt, Nabekura, and Dingwall is the same as provided in the rejection of claim 3 above.

37. **As to claim 8**, Baertsch as modified by Wilt and Nabekura does not explicitly disclose wherein said executing said commands from said stored sequence of commands is done at a different privileged mode level system.

38. However Dingwall teaches wherein said executing said commands from said stored sequence of commands is done at a different privileged mode level system (***i.e. Virtual Device Driver (VxD), 28, Fig. 2, run at most privileged level col. 3, lines 36-37***).

39. The motivation for further modifying Baertsch with the teachings of Wilt, Nabekura, and Dingwall is the same as provided in the rejection of claim 3 above.

40. **As to claim 9**, Baertsch as modified by Wilt and Nabekura does not explicitly disclose wherein said different privileged mode level is that of Interrupt Service Routine, whereby delay between the execution of successive commands is minimized.

41. However Dingwall teaches wherein said different privileged mode level is that of Interrupt Service Routine (***i.e. Virtual Device Driver (VxD), 28, Fig. 2, which is interrupt driven, runs at most privileged level col. 3, lines 36-38***), whereby delay between the execution of successive commands is minimized (***i.e. improves real-time response col. 2, line 49-50***).

42. The motivation for further modifying Baertsch with the teachings of Wilt, Nabekura, and Dingwall all is the same as provided in the rejection of claim 3 above.

43. **As to claim 10**, Baertsch as modified by Wilt and Nabekura does not explicitly disclose wherein said non real-time operating system is MICROSOFT WINDOWS™.

44. However Dingwall teaches wherein said non real-time operating system is MICROSOFT WINDOWS™ (***i.e. environment of WINDOWS™, col. 3, lines 33-34***).

45. The motivation for further modifying Baertsch with the teachings of Wilt, Nabekura, and Dingwall is the same as provided in the rejection of claim 3 above.

46. **As to claim 11**, Baertsch as modified by Wilt and Nabekura does not explicitly disclose wherein said sequence of commands process a same data set.

47. However Dingwall teaches wherein said sequence of commands process a same data set (***i.e. task needs to process data in buffer stored by audio/video device, col. 4, lines 59-60***).

48. The motivation for further modifying Baertsch with the teachings of Wilt, Nabekura, and Dingwall is the same as provided in the rejection of claim 3 above.

49. **As to claim 12**, Baertsch as modified by Wilt and Nabekura does not explicitly disclose wherein said same data set is a video camera image being captured and processed in real-time.

50. However Dingwall teaches wherein said same data set is a video camera image being captured and processed in real-time (***i.e. task needs to process data in buffer stored by audio/video device, col. 4, lines 59-60***)(***i.e. example task used to perform capture or playback of audio/video, col. 4, lines 5-6***).

51. The motivation for further modifying Baertsch with the teachings of Wilt, Nabekura, and Dingwall is the same as provided in the rejection of claim 3 above.

52. **As to claim 13**, Baertsch as modified by Wilt and Nabekura does not explicitly disclose wherein said providing said sequence of commands involves said commands being pushed one at a time into said sequence through a system call.

53. However Dingwall teaches wherein said providing said sequence of commands involves said commands being pushed one at a time into said sequence through a system call (***i.e. interrupt occurs which causes the processor to switch to VxD interrupt mode and execute RT interrupt handler 32, Fig. 2, col. 4, lines 51-23, RT interrupt handler 32, Fig. 2, wake up associated real-time task***).

54. The motivation for further modifying Baertsch with the teachings of Wilt, Nabekura, and Dingwall is the same as provided in the rejection of claim 3 above.

55. **As to claim 14**, Baertsch as modified by Wilt and Nabekura does not explicitly disclose wherein said storing said sequence of commands is performed through execution of a driver routine from a system file.

56. However Dingwall teaches wherein said storing said sequence of commands is performed through execution of a driver routine (***i.e. Virtual Device Driver***) from a system file (***i.e. Virtual Device Driver (VxD) is dynamic link library (DLL), col. 3, lines 33-36***).

57. The motivation for further modifying Baertsch with the teachings of Wilt, Nabekura, and Dingwall is the same as provided in the rejection of claim 3 above.

Response to Arguments

58. Applicant's arguments filed on April 4, 2011 have been fully considered but they are not persuasive. In response to the Final Office Action dated January 3, 2011 applicant argues in regards to claims 1-14:

> (1) Claim 1, as amended, clearly indicates that the sequence of commands, provided to the privileged mode and to be executed in real time, is provided to at least one CPU running the non real time operating system, and that the commands are initiated one at a time, by the at least one CPU, for execution. In contrast, Baertsch et aL describe storing the commands on a hardware device

(the Detector Framing Node (DFN)) and having the hardware device execute the sequence of commands in batch autonomously after the Host sends a "begin" command° This is evidenced by the passage found in column t4 at lines 48-54, where it states that "Event sequence 312 is initiated by sending a Begin Sequence command over computer communication bus 302. The extent of real-time control allotted to host computer 114 is confined to a determination of when event sequence 312 will begin. Subsequently, host computer 114 is completely removed from image acquisition". It is important to note the distinction between the DFN device driver and the DFN hardware device in Baertsch et al. The device driver is only a tunnel in the operating system to send the list of commands in advance and write them in the hardware for later execution. **Therefore, it should be understood that Baertsch et al. fail to teach or suggest at least the steps of "providing from said at least one application said sequence of commands to a privileged mode of said non real time operating system to be executed in real time" and "initiating one at a time, using the at least one CPU, execution of each of said commands from said stored sequence of commands"** (page 4, lines 1-16).

In response to argument (1), examiner respectfully disagrees and notes that Baertsch discloses providing from said at least one application said sequence of commands to a privileged mode of said non real time operating system to be executed in real time. Baertsch teaches *"Frame sequence 310 is then translated into event*

sequence 312 using a compiler, which knows the details of the target control hardware. Event sequence 312 is received by test control unit 311, then sent to DFN device driver 314 ", ***col. 14, lines 42-46***, which represents providing from said at least one application said sequence of commands to a privileged mode of said non real time operating system since "*test control unit 311" is part of "User App. 301", Figure 15*, and *" DFN device driver 314" operates at kernel mode, Figure 71* of *"host computer 114 running a non-real time operating system to support an event compiler",* ***col. 14, line 34****. "Once the event sequence 312 is known, the details are transmitted to DFN 304 for execution in real-time",* ***col. 14, lines 36-38,*** which represents to be executed in real time since *"Event sequence 312 is received by test control unit 311, then sent to DFN device driver 314, over computer communication bus 302, and finally to detector framing node 304. The event sequence 312 is then stored in preparation for execution",* ***col. 14, lines 44-48****.

In addition Applicant's arguments with respect to the limitation of "*initiating one at a time, using the at least one CPU, execution of each of said commands from said stored sequence of commands*" have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

59. Any inquiry concerning this communication or earlier communications from the examiner should be directed to KIMBLEANN VERDI whose telephone number is

(571)270-1654. The examiner can normally be reached on Monday-Thursday 7:30am-5:00pm EST..

60. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hyung Sough can be reached on (571)272-6799. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

61. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/S. Sough/
Supervisory Patent Examiner, Art Unit 2194
KV
June 1, 2011